The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

創作支援を目的とした類似度調節による ストーリー生成システムの構築

Story Creation System based on Sentence Similarity for Supporting Contents Creation

高橋 遼 *1　上野 未貴　井佐原 均
Haruka Takahashi　Miki Ueno　Isahara Hitoshi

*1豊橋技術科学大学
Toyohashi University of Technology

In recent years, there are lots of researches to generate creations such as comics and novels by computing methods. Especially, the orders of sentences is very important for the quality of stories. Thus, the aim of the research is to construct story creation system in order to consider the flow of the story based on the sentence similarity.

1. はじめに

近年，計算機の性能の向上によるニューラルネットワークを用いた学習が盛んである．その一つとして，漫画や小説，アニメーションといった人間の創作物を計算機によって理解したり生成する研究が広がりつつある [1]．これらの研究から，人間が創作活動を行う際の補助を行うことも可能であると考えられる．

創作者はテキスト，画像や音声などの情報を工夫して組合せてストーリーを表現する [2]．ストーリーは創作物の根幹であり，カテゴリ，評価や独自性に大きく影響する．そのため，ストーリーを作成する過程を計算機上で処理することは挑戦的な課題である．

本研究では，読者が違和感を感じにくいストーリー構成の分析や，文の順序を変更して新しい作品の生成へ応用できるか調べるため，物語のストーリー自動生成システムを構築する．

2. 先行研究

2.1 ストーリー創作支援

ストーリー創作支援の従来研究としては，質問集合とグラフに基づく物語全体の流れを管理可能な創作支援システムの提案 [3] がある．この研究では，創作支援システムを作成し，創作支援システムへの入力から創作過程の解析と創作支援をする．あらかじめ準備されたカテゴリから創作者が今回考案するストーリーに適したものを選び，フォームを埋める形式でストーリープロットを作成することができる．しかし，ストーリーの生成は創作者によるもので，計算機が自動生成はしていない．

2.2 文自動生成

文の自動生成の従来の研究としては，単語の分散表現を用いた意味予測に基づく雑談応答生成 [4] がある．この研究では Recurrent Neural Network(RNN) を用いて, Twitter 上の会話文を学習させ，ある会話文の入力に対して，適切な応答文の作成を行っている．従来の One-hot 表現を用いた方法と違い，単語の分散表現を入力としている．従来の方法よりも多様性の面では向上したが，適切性に関しては大きく向上しなかった．また，文法的な誤りがある文も生成された．

この研究は，1 文に対して 1 文を返す対話文を対象としたものであり，ストーリーといった複数文の出力には対応していない．

3. ストーリー生成システム

本研究では分散表現を利用し，文間の類似度に基づくストーリー生成システムを構築する．

3.1 コーパスの作成

コーパスには，3 つのデータを用いる．用いたデータを以下に示す．

1. Manga109 の 4 コマ漫画のセリフデータ（ 5145 文）
2. 青空文庫のあらすじ（ 7439 文）
3. インターネット上の小説投稿サイトから集めた文章（ 383494 文）

本稿ではこの中でも最もデータ数が多く，結果が良かった小説投稿サイトの文章を利用する．「インターネット上の小説投稿サイト」のデータはインターネットからクローリングして収集した．これらを句点で分割し，1 文区切りにしたものを文集合として利用する．

3.2 Paragraph Veector のモデル作成

Paragraph Vector[5] のモデルを作成する. Paragraph Vector とは文を分散表現に変換する技術である. 分散表現に変換することで, コサイン類似度などを用いて, 文同士の類似度を計算できるようになる.

Paragraph Vector の学習に使用する文から単語を抽出する. 文に単純な分かち書きしたものを学習に使用すると, 文の意味への影響が小さい助詞や助動詞, 記号等も学習に使用される. そこで, 形態素解析をし, 文の意味に関連しにくいと考えられる品詞の単語を除去する. 形態素解析には MeCab を用いた. また, MeCab の辞書には NAIST Japanese Dictionary*1 を用いる. 今回学習に使用する品詞は名詞, 動詞, 形容詞, 副詞, 連体詞である.

連絡先: 高橋 遼，豊橋技術科学大学，〒 441-8580　愛知県豊橋市天伯町雲雀ケ丘 1 − 1，takahashi@lang.cs.tut.ac.jp

*1　NAIST Japanese Dictionary, http://naist-jdic.osdn.jp/

表 1: Doc2Vec のパラメータ

手法	次元サイズ	単語の最小登場回数	エポック数
DM-PV	100	1	100

DM-PV: Distributed Memory Paragraph Vector[6]

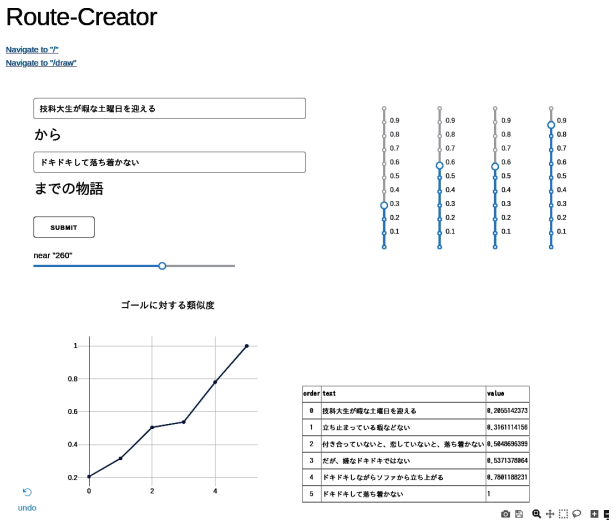

図 1: ストーリー生成システムのユーザインタフェース

Paragraph Vector を用いるためには Python 製のソフトウェアである Gensim[*2] を用いる．Gensim の中の Paragraph Vector を用いることができるライブラリである Doc2Vec を用いる．表 1 に Doc2Vec のパラメータを示す．

3.3 ルート作成

上述の通り作成したモデルを用いて，ストーリーを作成する．システムの入力はストーリーの「スタートの文 r_s」と「ゴールの文 r_g」であり，スタートからゴールに対して，類似度が高くなるようにストーリーの遷移経路を以下の手順で作成する．

1) 空のキュー $\mathbf{R}_s$ を用意する．ゴールの文に最も近い位置にモデル内で存在する文を仮のゴールの文 r'_g とする．$\mathbf{R}_s$ に r_s を追加する．
2) $\mathbf{R}_s$ の末尾の文に対し近傍 n の文を取得し，r'_g との類似度の高い順にソートした配列 $\mathbf{A}_s = \{a_1, a_2, ..., a_n\}$ を作成する．
3) $\mathbf{R}_s$ に既存でないという条件を満たす，$\mathbf{A}_s$ のインデックス番号が最も小さい文 a_i を $\mathbf{R}_s$ に追加する．
4) $\mathbf{A}_s$ に r'_g が存在すれば，r'_g と r_g を $\mathbf{R}_s$ に追加しルート作成は終了する．終了しない場合，2) に戻って処理を繰り返す．

作成された $\mathbf{R}_s$ がストーリーとなる．

3.4 ユーザインタフェースの実装

本研究は創作支援を目的としており，将来的にユーザが使用することを想定し，WEB アプリケーションにてストーリー生成システムの実装を行った．今回，開発に使用した言語は Python であり，用いた WEB アプリケーション開発用ライブラリは Dash by plotly [*3] である．図 1 に作成したシステムのユーザインタフェースを示す．図 1 の左上が入力するためのテキストボックスである．右上の類似度調節機能でストーリー内の類似度の遷移を決定する．作成されたストーリーは右下に表示される．左下にはストーリーの類似度の遷移の軌跡がグラフとして表示される．

*2 Gensim, https://radimrehurek.com/gensim/
*3 Dash by plotly, https://plot.ly/products/dash/

表 2: 生成されたストーリーの例 1

	文	類似度
0	技科大生が暇な土曜日を迎える	0.206
1	立ち止まっている暇などない	0.316
2	付き合っていないと、恋していないと、落ち着かない	0.505
3	だが、嫌なドキドキではない	0.537
4	ドキドキしながらソファから立ち上がる	0.780
5	ドキドキして落ち着かない	1.00

表 3: 実験 1 のアンケート結果

入力	近傍	文数	自然に感じた	ほとんど自然に感じた	やや不自然に感じた	不自然に感じた
1	50	9	0	1	2	1
	100	7	0	1	2	1
	300	7	0	2	2	0
2	50	29	0	0	1	3
	100	9	0	1	2	1
	300	11	0	0	4	0
3	50	7	0	0	3	1
	100	6	1	1	2	0
	300	6	0	0	2	2
4	50	7	1	2	1	0
	100	9	1	2	1	0
	300	4	2	1	1	0

3.5 生成結果

表 2 に生成されたストーリーの例を示す．

生成結果は，文数をスタートとゴールの文を含めた 6 文に固定し，類似度を図 1 の右上のスライダーのように調節した．

4. 実験 1

4.1 アンケート評価

本システムの評価を行うためにまず，アンケートをした．文数を固定せず，類似度の調節無しとした．スタートからゴールに到達するまでに要した遷移回数が文数となる．スタートとゴールのペアが異なる 4 パターンの入力に対し，それぞれ近傍数を 3 種類： 50, 100, 300 に変化させ，ストーリーを生成した．そして，それらのストーリーに対し， 4 人のユーザに 4 段階 : 自然に感じた，ほとんど自然に感じた，やや不自然に感じた，不自然に感じたのいずれかを回答して評価をしてもらった．表 3 にアンケートの結果として各入力に対する回答数を示す．

4.2 ランダムとの比較評価

次に，ランダム生成したストーリーとシステムを用いて作成したストーリー対 8 パターンを 4 人のユーザに比較をしてもらった．方法は，各入力パターンに対し，スタートの文からゴールの文までをつなぐ 5 文と 10 文をそれぞれシステムを用いる方法とコーパスから無作為に抽出する方法で決定し，スタートの文とゴールの文と合わせてストーリーとする．生成されたストーリーを比較し，どちらがより自然に見えるストーリーかをユーザに選んでもらう．比較は，文数が同じもの同士とした．4.1 のアンケート評価により，最も不自然と感じるユーザが少なかった近傍数を 300 に固定した．また，図 2, 図 3 はシステムの 5 文と 10 文の場合の類似度調節機能のス

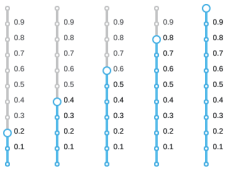

図 2: 類似度調節 5 文

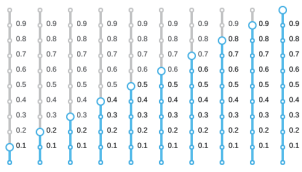

図 3: 類似度調節 10 文

表 4: ランダムとの比較結果

パターン	5 文		10 文	
	システム	ランダム	システム	ランダム
1	3	1	4	0
2	3	1	3	1
3	4	0	4	0
4	4	0	3	1
5	3	1	4	0
6	3	1	4	0
7	4	0	4	0
8	2	2	4	0

ライダーの設定である．類似度調節機能のスライダーはストーリーに意図的なアクセントが発生しないようにした．

表 4 に比較結果を示す．表 4 内の各パターンに対する行の数字は， 4 人中何人が選択したかを示している．

4.3 考察

アンケートによる評価では，システムによって出力されるストーリーでは取得する近傍の文数を大きくすることで，不自然と感じる人が減少することがわかった．

ランダムとの比較では，5 文と 10 文のどちらの場合においても，システムは優位性を示すことができた．しかし，ランダムなストーリー生成であっても 5 文からなるストーリーでは自然と感じられる場合が見られた．一方で, 10 文の場合では 4 人のユーザがほとんどの入力パターンにおいてシステムの方が自然であると回答した．本システムでは, 文の類似度が近い文がつながるため, 文の数が増加しても全体の名詞種数が多くなりにくく, システムを用いることで，5 文以上の文数では自然なストーリーが生成されやすくなると考えられる．

ストーリーが自然と感じられなかった原因としては，文の数や長文の影響が考えられる．文の数や長文が含まれると，ストーリー全体に含まれる固有名詞の数が増える．これが，ストーリー内での場面や時代，世界観に矛盾を生み出す原因となる．対策としては，固有名詞等を代名詞に変換することや，文数を少なく固定して生成すること，文長を制限し，長文の登場を防ぐことで改善できると考えられる．また，コーパスサイズにより取得される類似文の類似度の平均が大きく変わることが考えられるため，コーパスサイズに応じた適切な近傍の値を用意することで改善が期待できる．

また，文によっては同じ意味の文が複数出現し，それらがモデル内に多数存在し密集することで，類似文があまりに似通った文しか取れないという問題がある．これは，ストーリーの遷移ができず，同じような文ばかりが連続する原因となる．対策としては，極度な類似文はモデル作成時に数を減らすことや近傍 n を一時的に大きくすることが考えられる．

5. コーパスの前処理の変更

実験 1 では，分散表現より類似文を取得する際に，長い文は取得した文との類似度が高くなく，実際に読んでも類似しているとは言い難い文も取得されていた．

そこで，実験 1 で用いた文の処理方法とは異なる方法で文を処理し，コーパスを作成した．表 5 に変更点を示す．3.1 で準備したインターネット上の小説投稿サイトの作品は文体が自由に書かれており，全角記号の出現数が多い．また，文末と思われる点でも句点が用いられていない場合があり，その場合空白や改行を文の終端として用いている場合がみられた．以前のコーパスでは，鉤括弧の前後をつなげて一文としていた．しかし，鉤括弧内の文を独立して抜き出すために，分割することにした．

表 5: 前処理の変更点

対象	変更前	変更後
空白	削除	文の終端として，そこで文章を分割
鉤括弧 (「」)	読点に置換	文の終端として，そこで文章を分割
全角記号	処理なし	感嘆符 (！)，疑問符 (？)，二重鉤括弧 (『』) 以外を削除

学習データ量は，前処理の変更により，383494 文から 663066 文に増加した．

6. 実験 2

実験 1 の結果から，文の長さが長いと違和感を覚えやすいという結果があり，これらを改善することで，生成されるストーリーの自然さに改善がみられないかを 4.1 のアンケート評価で比較実験を行う．ランダムとの比較は，4.2 の時点で既に優位性がみられたため取り組まない．

6.1 実験方法

5. で変更したコーパスを用いたストーリー生成の実験をする．3.5 と同様のパラメータで，前処理変更後のモデルに対してもストーリーを生成させた．また，実験 1 と同様に文数を固定せず，類似度の調節無しで行った結果を用いる．入力文も同様のものを用いる．

6.2 結果と考察

表 6 に生成されたストーリーの例を示す．表 7 にアンケートの結果を示す．

前処理を変更したコーパスを用いた実験では,「やや不自然に感じた」,「不自然に感じた」と答える人がわずかに減少した．これは実験 1 で述べた長文が自然さに影響するという点において，コーパスの文がより細かく分割されたことにより，長文の割合が減ったため改善されたと考えられる．また， 1 文が短くなったことにより， Paragraph Vector によって，文の分散表現を獲得し，類似度の計算を行う際に，文の長さが極端に違うものが減少しているため，より正確に類似度を測ることができるようになったと考えられる．

しかし，表 7 の入力 4 に関しては，表 3 の結果よりも変更後のコーパスの方が「自然に感じた」と回答したユーザが少なくなった．表 3 の入力 4 の結果では文が長いものも含まれていたが，固有名詞が少なく，また「俺」や「彼女」といった代名詞の使用が多い．これをユーザーがうまく補完して，話のつながりを自然と捉えられたと考えられる．対して，表 7 の結果では，文数が増えたこともしくは類似度の計算が改善して類似文を集めやすくなったことの影響からか，同じ表現を何度も繰り返すようなストーリーとなってしまった．類似文上位 n 件を取得する現在の方法では，新しいコーパスを用いると 4.3 で述べた同じ意味の文が複数出現する問題に関しては悪化させる可能性がわかった．

表 6: 生成されたストーリーの例 2

	文	類似度
0	技科大生が暇な土曜日を迎える	-0.013
1	立ち止まっている暇などない	0.338
2	落ち着かないか、[キャラクター名]	0.541
3	心の中が落ち着かない	0.568
4	なんだろう、落ち着かない	0.618
5	ドキドキして落ち着かない	1.00

表 7: 実験 2 のアンケート結果

入力	近傍	文数	自然に感じた	ほとんど自然に感じた	やや不自然に感じた	不自然に感じた
1	50	7	0	2	2	0
	100	6	0	3	1	0
	300	5	0	0	3	1
2	50	10	0	0	2	2
	100	10	0	3	1	0
	300	7	2	1	1	0
3	50	10	0	1	1	2
	100	6	0	2	2	0
	300	6	1	1	2	0
4	50	10	0	0	2	2
	100	10	0	0	2	2
	300	5	0	2	1	1

上記のことを踏まえ，今後は付近の文との距離や数に影響されにくくするために，従来のストーリーが場面に応じてどの程度のベクトル変化量があるかを解析し，文を選択していくよう改良する必要があると考えられる．

表 2 と表 6 に示されるように，前後の文に同様の単語が出現する．これは 4.3 に問題点として挙げた同じ意味の文が複数出現するという問題にも近いが，用いている Paragraph Vector が単語に影響を受けやすいことが影響している．「〜ない」といった反意的な表現は学習時に直接的には考慮されないため，肯定文と否定文であっても出現する単語が似ている場合に連続する文として選択される場合がある．「〜する」と「〜しない」といった単語はモデル学習時には距離が遠くなるような学習を行う必要があると考えられる．そこで，反意を重視して特徴量の調整をしたり，例えば編集距離といった他の手法と組み合わせて文の類似度を算出することが有効ではないかと考えられる．

7. 創作支援への展望

本研究では，異なる 2 つの入力文に対して，それらを繋ぐストーリーを作成した．表 2，表 6 の「暇」と「落ち着かない」のようにほとんど対極とも取れる状態を結び，その間を補完するストーリーが作成できる．また，この入力が人間の目から見て明らかに繋がりにかける文であったとしても，少々強引になるかもしれないが，類似度調節を利用して段階的に結ぶことが可能である．よって，これまでの人の創作者の発想では思いつかなかった作品が生まれるきっかけになり得ると考えられる．

また，創作者の過去の十分な量の複数作品を学習することで，作者の過去に用いた文を参考に新たなストーリーを生み出すことが可能である．小説から小説でなくとも，小説作品のデータから，歌詞や短編の作品といった違う分野に活用することも考えられる．

また本研究を応用することで，作品のあらすじとして複数文を入力して文間の類似度の変化を得て，ストーリーのスタートからゴールまでの変化をパターンに分類し解析することも視野に入れている．

従って，本研究は創作支援に貢献できると考えられる．

8. まとめ

本研究では，創作支援を目的としてストーリー生成システムを構築した．文を類似度に基づいて順序変更することで人が違和感を覚えないストーリーを生成できる可能性を示した．また，以下のことがわかった．

- 文数が増えたり長文が含まれると，文中に含まれる固有名詞等の出現回数が増え，ストーリーの自然さが損なわれる
- 文中の固有名詞を代名詞に変換する等の対策を取り，改善する必要がある
- 記号の除去や文を分割する箇所を変更することで生成されるストーリーの結果が改善する

特にストーリーという観点で同じ単語を含まないが同様の意味の文は距離が近く，文に登場する単語は似ていても意味が反する文は遠くに置かれるモデルの作成が望まれる．また，コーパスに含まれる類似文の数がモデル内の距離の変化量に大きく影響するため一定距離での変化が可能なように改良が必要である．

謝辞

本研究は，一部 JSPS 科研費 (グラント番号:JP17K17809) および JST, ACT-I(グランド番号:JPMJPR17U4) の支援による．

参考文献

[1] 上野 未貴, ”創作者と人工知能: 共作実現に向けた創作過程とメタデータ付与 4 コマ漫画ストーリーデータセット構築”, 人工知能学会, 4Pin1-16 (2018)

[2] 福田 清人, 上野 未貴, 藤野 紗耶, 森 直樹, 松本 啓之亮, ”ストーリーの自動生成を目的としたストーリーモデルの提案”, 言語処理学会 P6-1 (2016)

[3] 葛井 健文, 上野 未貴, 井佐原 均, ”質問集合とグラフに基づく物語全体の流れを管理可能な創作支援システムの提案”, 人工知能学会, 2E2-04, (2017)

[4] 古舞 千暁, 滝口 哲也, 有木 康雄, ”単語の分散表現を用いた意味予測に基づく雑談応答生成”, 日本音響学会, ROMBUNNO.2-Q-12, (2018)

[5] Quoc Le, Tomas Mikolov, ”Distributed Representations of Sentences and Documents”, Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043, (2014)

[6] Andrew M. Dai, Christopher Olah, Quoc V. Le, ”Document Embedding with Paragraph Vectors”, arXiv:1507.07998, (2015)

[7] Y.Matsui, K.Ito, Y.Aramaki, A.Fujimoto, T.Ogawa, T.Yamasaki, K.Aizawa, ”Sketch-based Manga Retrieval using Manga109 Dataset”, Multimedia Tools and Applications, Springer, (2017)